CAMEO OCL TRANFORMER FOR MAGICDRAW™

USER MANUAL

Date: October 2012

Version: 1.2

All contained material is considered proprietary information owned by Nomoseire Limited and is not to be shared, copied, or reproduced.

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

Table of Contents

1 Introduction 3
1.1 Usage 3
2 Getting Started 4
2.1 Install Plugin 4
2.1.1 Install Java JDK if None Available 4
2.1.2 Point MagicDraw to Java JDK 4
2.2 Getting Started with an Example Project 5
2.2.1 Open the BulkPayments Project from the Example Directory 5
2.2.2 What's in the Example Project 6
2.2.3 What you can do with the Project 7
2.2.4 Look at How the Project is Configured 7
2.2.5 Generate a Ruleset 8
2.2.6 Execute the Ruleset 9
3 Working with Your Own Project 12
3.1 Configure the project 12
3.1.1 Including the Cameo_OCL_Transformer module 12
3.2 Generate an XML schema for the Class Model 14
3.3 Configure the Class Model Package 14
3.3.1 Configuring class model package element 14
3.4 Adding OCL Rules to Model 16
3.5 Generate a Ruleset 19
4 Executing Rulesets via MagicDraw 20
4.1.1 Create Example XML Files 20
4.1.2 Validate XML 20
5 Learn More 21
6 Contacting Nomos Software Support 22

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

Table of Figures

Figure 1 Point MagicDraw to Java JDK 5
Figure 2 Open Project 6
Figure 3 Example Project 6
Figure 4 Configured Package 7
Figure 5 Generate Ruleset 8
Figure 6 Generation Errors 8
Figure 7 Example Payments XML File 9
Figure 8 Validate XML 9
Figure 9 Select XML File 10
Figure 10 Select Directory containing Generated Rules 11
Figure 11 Results of 'Validate XML' 11
Figure 12 Use Module Menu Item 12
Figure 13 Use Module Dialog 13
Figure 14 Cameo_OCL_Transformer Module shown in Tree 13
Figure 15 Stereotyping a Package 15
Figure 16 Setting the Ruleset and XMLSchema Properties 16
Figure 17 OCL Rules in Tree 17
Figure 18 OCL Rule Dialog 18
Figure 19 Generate Ruleset 19
Figure 20 Regenerate Ruleset 19
Figure 21 Validate XML Menu Item 20
Figure 22 Select Directory containing Generated Rules 21
Figure 23 ‘Validate XML’ Results 21

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

1 Introduction

The Cameo OCL Transformer generates executable java rulesets from OCL captured in MagicDraw™ models.

The java rulesets can be executed over XML data that matches the model. That is, the rulesets execute over XML data that is structured according to an XML schema that maps to and is generated from the UML model in which the OCL expressions are captured.

The Cameo OCL Transformer product is powered by the RuleX product from Nomos Software.

1.1 Usage

With the Cameo OCL Transformer, MagicDraw modellers can:

- Add OCL rules to class models
- Generate XML Schemas from class models (using native MagicDraw functionality)
- Generate java rulesets based on the OCL
- Validate XML files against the XML Schema and the ruleset from the MagicDraw user interface
- Deploy the rulesets outside the MagicDraw environment, and validate XML files against the XML schema and the ruleset via a programmatic interface

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

2 Getting Started

2.1 Install Plugin

You will need:

- MagicDraw™ UML, minimum version 17.0.2
- Cameo Data Modeller, minimum version 17.0.2
- Java JDK version 6.0 or later

The plugin works on Windows, UNIX and Linux.

Follow the standard MagicDraw procedure for installing plugins:

- Obtain a license for the Cameo OCL Transformer Plugin
- Activate the license and the plugin will be installed automatically

MagicDraw can be installed with a java JRE or a java JDK. For the OCL Transformer, MagicDraw needs to use a java JDK (Java Development Kit) rather than a JRE (Java Runtime Environment).

If MagicDraw has been installed with a JRE rather than a JDK, you will need to:
- install a JDK, if none is installed already
- point MagicDraw to a JDK

If MagicDraw is configured to use a JRE rather than a JDK, you will get the following error when you try to use the Cameo OCL Transformer plugin.

```
"Fatal Error, no java compiler found. Please ensure JAVA_HOME is
pointing at a JDK and not a JRE"
```

2.1.1 Install Java JDK if None Available

If a Java JDK, version 6.0 or later, is installed already, skip this step.

Java JDK is usually installed in C:\Program Files\Java.

If you need to install a Java JDK, go to http://www.oracle.com/technetwork/java/javase/downloads/index.html, download the most recent Java JDK and install.

2.1.2 Point MagicDraw to Java JDK

Update the JAVA_HOME property for MagicDraw to point to a JDK rather than a JRE:

- Identify the location and version of the installed JDK
  - usually found under C:\Program Files\Java
- Locate the mduml.properties file under <MagicDraw_install_directory>/bin

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

- Open the file e.g. in notepad, and set JAVA_HOME to the location of the installed JDK
    - E.g. JAVA_HOME=C\:\\Program Files\\Java\\jdk1.6.0_19

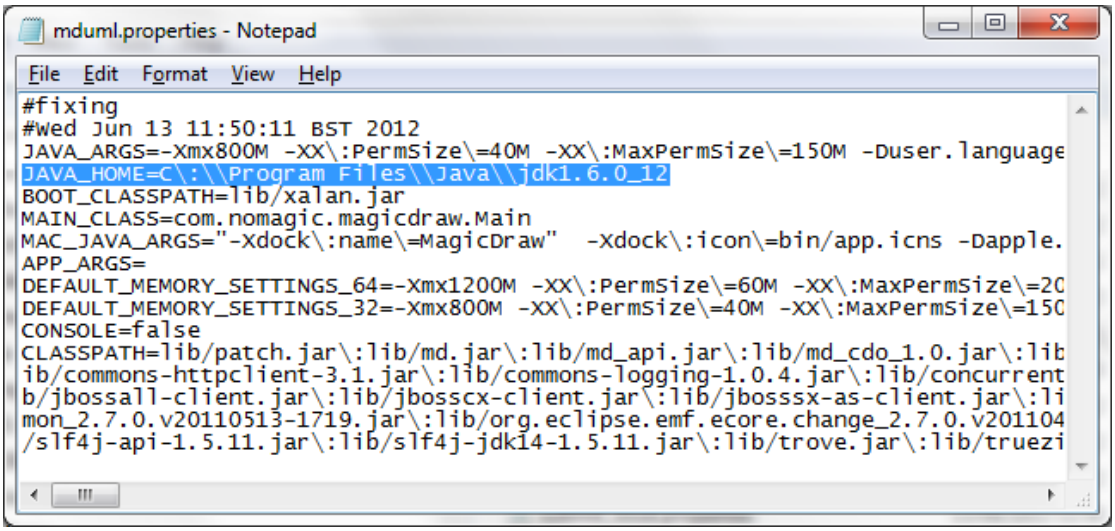

Figure 1 Point MagicDraw to Java JDK

2.2 Getting Started with an Example Project

Understand how to use the Cameo OCL Transformer by looking at an example project.

The plugin is shipped with an example BulkPayments project located at:
<MagicDraw_install_directory>/samples/CameoOclTransformer

Check for example projects:

- Under Downloads on the NoMagic site at: http://www.nomagic.com/products/magicdraw-addons/cameo-ocl-transformer.html
- And under MagicDraw Resources on the Nomos Software site at:http://nomos-software.com/resources

2.2.1 Open the BulkPayments Project from the Example Directory

1. Select File->Open Project…
2. Browse to samples/CameoOclTransformer/bulkpayments
3. Select BulkPayments.mdzip and open

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

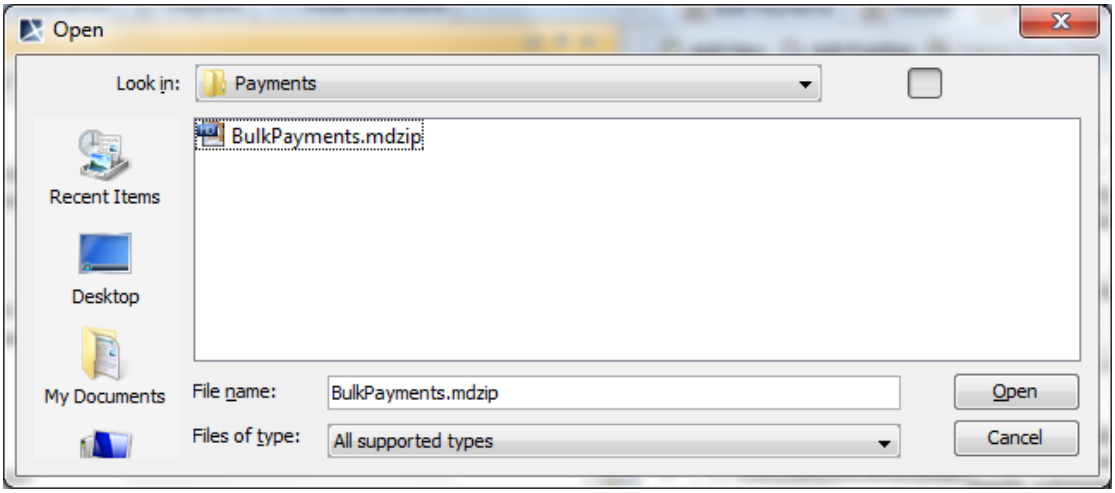

Figure 2 Open Project

2.2.2 What's in the Example Project

When you open the project you will see:

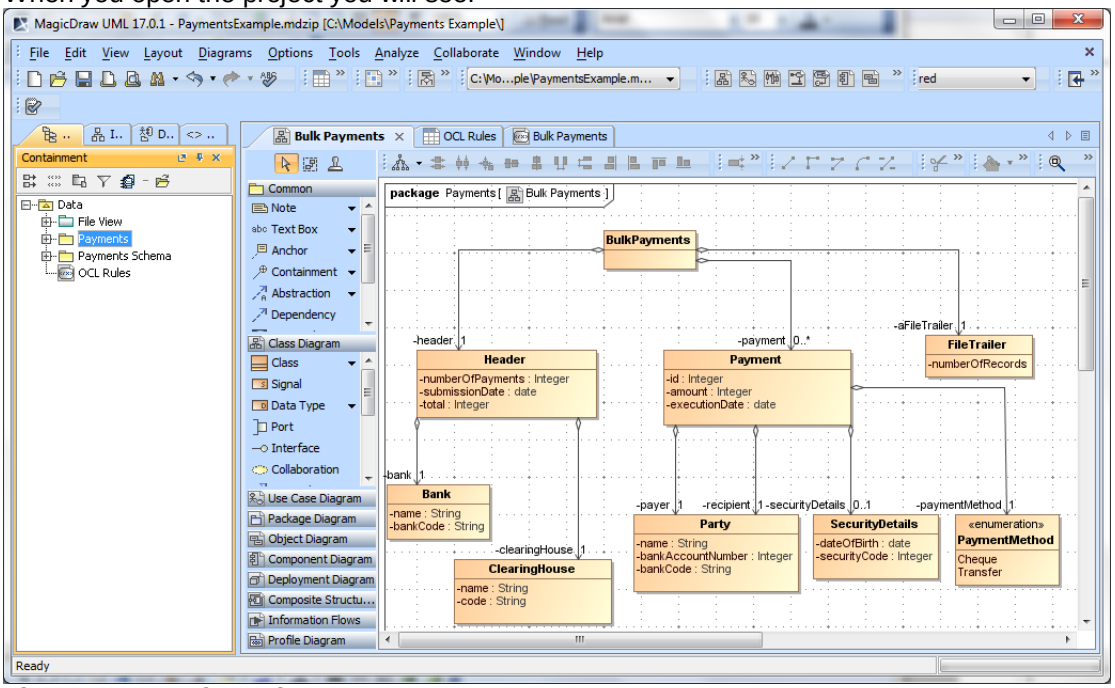

Figure 3 Example Project

Under the 'Data' element in the tree, you will see:

- 'File View' package containing paymentsSchema.xsd
- 'Payments' Package
- 'Payments Schema' Package
- An 'OCL Rules' table

Click through the tabs on the righthand pane. You will see:

- a diagram called 'Bulk Payments' of the class model from the 'Payments' package
- a table of OCL rules
- a diagram called 'Bulk Payments' of the class model transformed to an XML schema model from the 'PaymentsSchema' package *(note: this diagram will not be visible if the Cameo Data Modeler is not installed)*

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

2.2.3 What you can do with the Project

With this project you can:

- Generate an XML Schema
- Generate an executable ruleset based on the OCL
- Execute the rules over example XML files

An XML Schema, paymentsSchema.xsd, has already been generated and is available in:

<MagicDraw_install_directory>/samples/CameoOclTransformer/bulkpayments

Example XML files are provided in:

<MagicDraw_install_directory>/samples/CameoOclTransformer/bulkpayments/xmlfiles

For instructions on how to generate the XML Schema, take a look at http://nomos-software.com/blog.

2.2.4 Look at How the Project is Configured

Double click on the Payments package element. In the specification dialog that opens you will see:

- a Ruleset name, “payments”

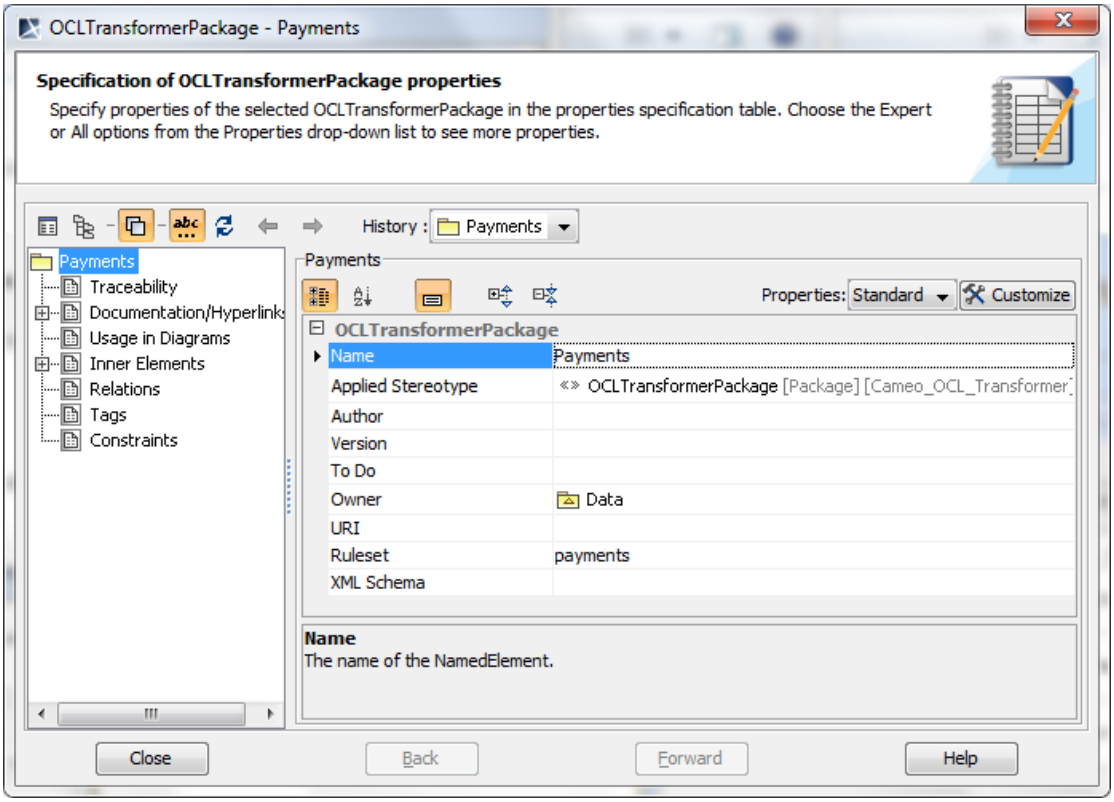

Figure 4 Configured Package

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

Note that the package is stereotyped as an OCLTransformerPackage. When you generate a ruleset, the OCL rules associated with the ruleset in this package will be generated and compiled against the XML schema located in the same directory as the project.

2.2.5 Generate a Ruleset

Select the Payments package, right click and select Cameo OCL Transformer->Generate Ruleset.

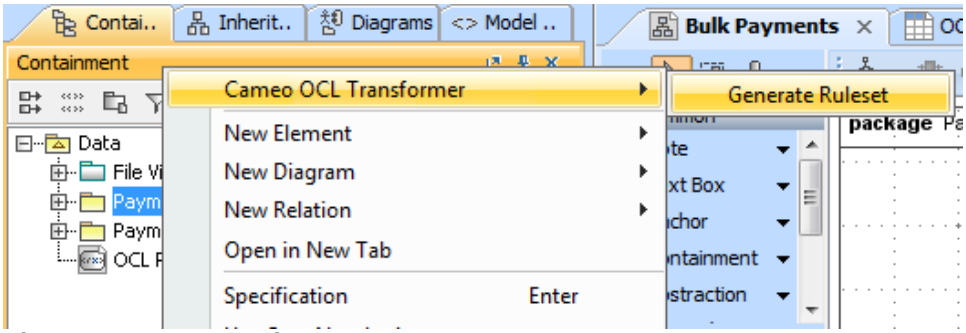

Figure 5 Generate Ruleset

Note: If you receive an error relating to the java compiler and JAVA_HOME at this stage, resolve by following steps outlined in the install procedure.

An error will be detected in one of the rules and will be displayed in a Validation Results pane at the bottom of the window.

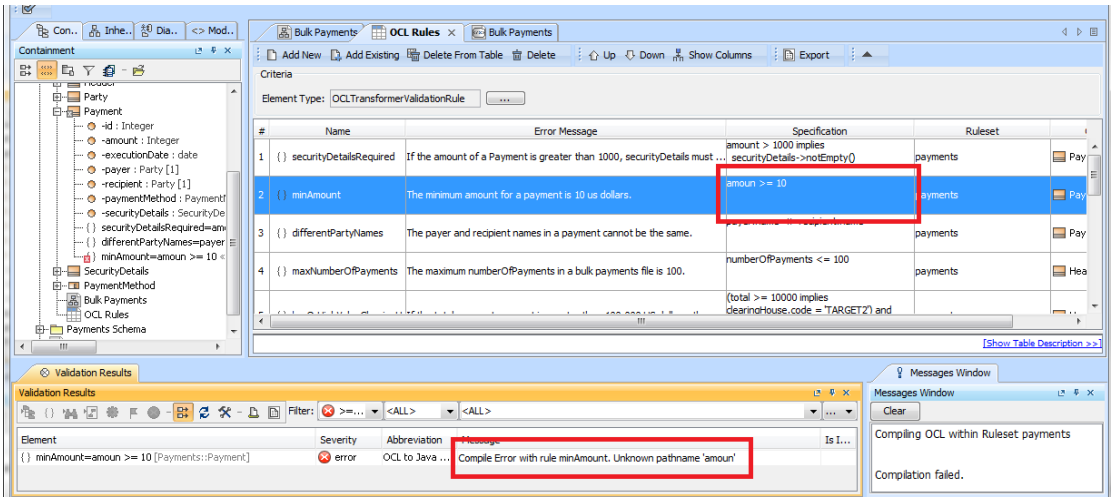

Figure 6 Generation Errors

The problem is that the rule refers to an **amoun** property on the Payment class. But the property is actually named **amount.** To correct, click on the OCL Rules tab, find the 'minAmount' rule in the table of OCL rules (highlighted above), click on the OCL and change **amoun** to **amount.**

Generate the ruleset once more, and all rules should compile.

Tip: You can regenerate the ruleset by clicking on the Refresh button on the Validation Results pane.

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

2.2.6 Execute the Ruleset

You can now validate XML files against the ruleset. You can use the example payments XML files provided in:

<MagicDraw_install_directory>/samples/CameoOclTransformer/bulkpayments/xmlfiles

These files are valid according to the XML schema generated from the MagicDraw model. Some of the XML files pass ruleset validation, while others fail ruleset validation.

Here is what one of the example XML files looks like.

```
<n:bulkPayments xmlns:n="Payments" xmlns:xsi="http://www.w3.org/2001/
  <payment>
    <id>0</id>
    <amount>20</amount>
    <executionDate>2001-12-17T09:30:47.0Z</executionDate>
    <payer>
      <name>aPerson</name>
      <bankAccountNumber>0</bankAccountNumber>
      <bankCode>BIC45678</bankCode>
    </payer>
    <recipient>
      <name>aPerson</name>
      <bankAccountNumber>0</bankAccountNumber>
      <bankCode>String</bankCode>
    </recipient>
    <paymentMethod>Cheque</paymentMethod>
    <securityDetails>
      <dateOfBirth>2001-12-17T09:30:47.0Z</dateOfBirth>
      <securityCode>0</securityCode>
    </securityDetails>
  </payment>
  <header>
    <numberOfPayments>1</numberOfPayments>
    <submissionDate>2001-12-17T09:30:47.0Z</submissionDate>
    <total>20</total>
    <bank>
      <name>String</name>
      <bankCode>BIC45678</bankCode>
    </bank>
    <clearingHouse>
      <name>String</name>
      <code>STEP2</code>
    </clearingHouse>
  </header>
  <aFileTrailer>
    <numberOfRecords>1</numberOfRecords>
  </aFileTrailer>
</n:bulkPayments>
```

Figure 7 Example Payments XML File

To validate an XML file, select Tools->Cameo OCL Transformer->Validate XML…

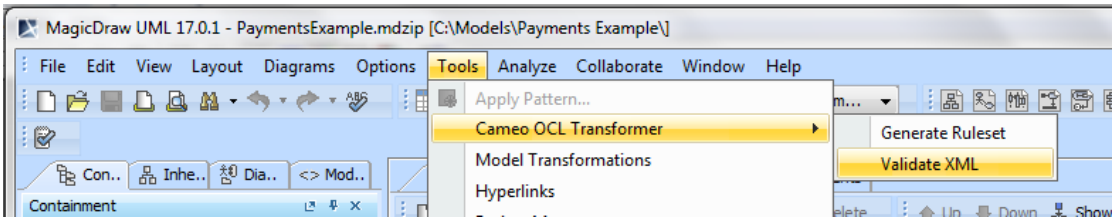

Figure 8 Validate XML

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

You will be prompted to select an XML file

- Browse to <MagicDraw_install_directory>/samples/CameoOclTransformer/bulkpayments/xmlfiles
- Select one of the XML files

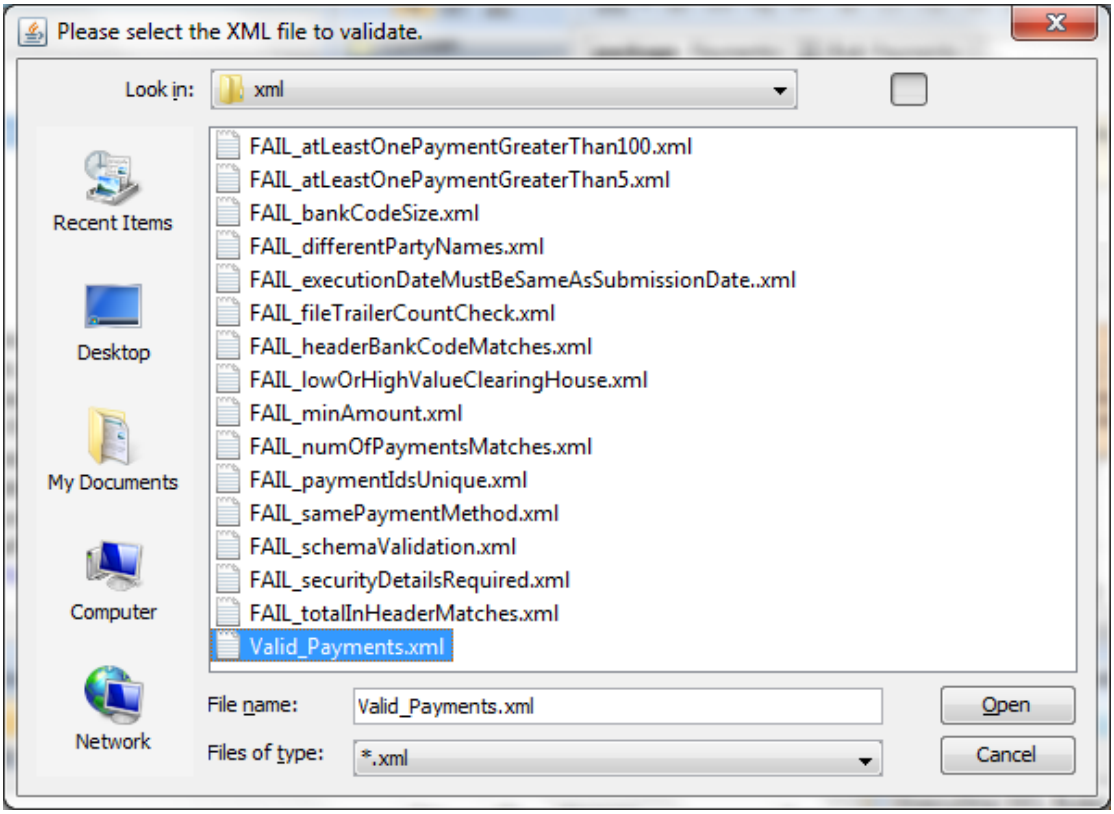

Figure 9 Select XML File

You will be prompted to select the directory containing the generated rules. The directory name is the same as the name of the project:

- Select PaymentsExample

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

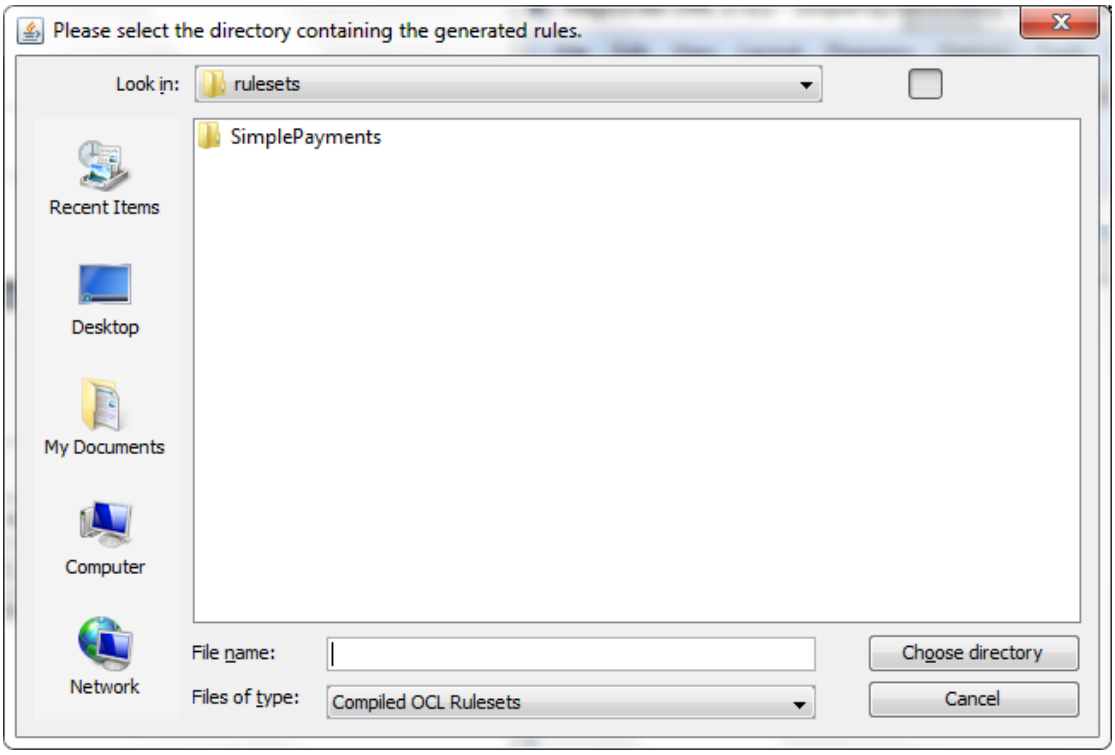

Figure 10 Select Directory containing Generated Rules

The XML file will be validated against the generated rules.

Any errors detected in the file are displayed in the 'Validation Results' area.

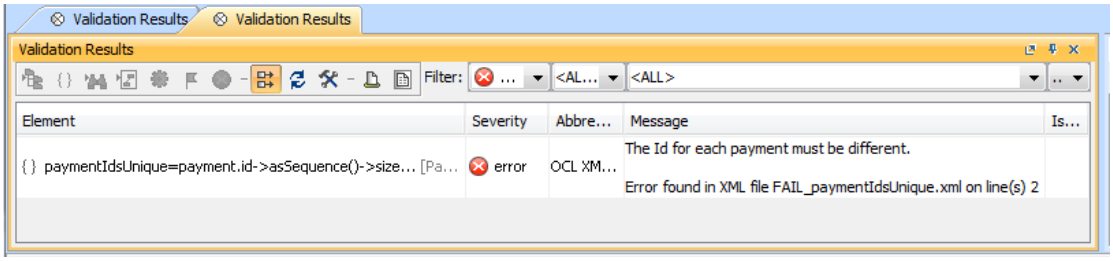

Figure 11 Results of 'Validate XML'

You have now validated an example file against the ruleset manually. The ruleset can also be called programmatically by making calls directly to a jar (java archive). For details, see the document "Nomos Software Using the Runtime Java API.doc". This can be found in:

<MagicDraw_install_directory>/plugins/com.nomagic.magicdraw.cameoocltransformer

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

3 Working with Your Own Project

The OCL Transformer is used to generate rulesets from OCL contraints on class models.

You will need a project containing a class model.

To use the OCL Transformer:

- Configure the project for the OCL Transformer
- Generate an XML Schema from the class model
- Configure the class model package for the OCL Transformer
- Add OCL Contraints to the class model

You will then be able to:

- Generate rulesets from the model
- Execute the rulesets against sample XML files that you create
- Deploy the ruleset

3.1 Configure the project

You will need to:

- Include the Cameo_OCL_Transformer module in the project

3.1.1 Including the Cameo_OCL_Transformer module

To include the Cameo_OCL_Transformer module in the project:

- Select File->Use Module
- Locate the Cameo_OCL_Transformer module under the <install.root>\profiles
- Click Finish (using the default Module Settings)

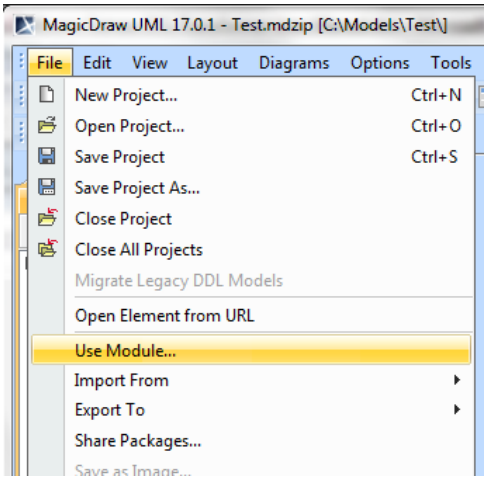

Figure 12 Use Module Menu Item

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

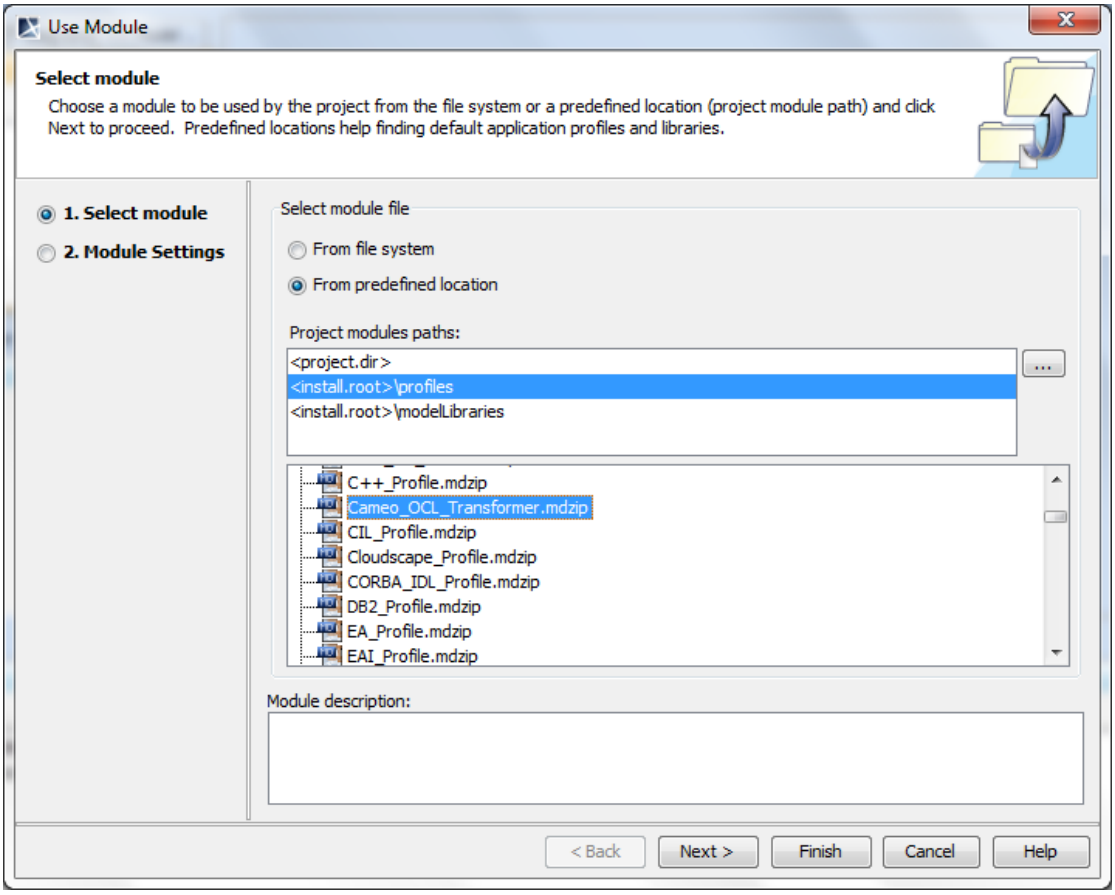

Figure 13 Use Module Dialog

You may be prompted to choose whether to display the modules in the tree. If you choose to display the module in the tree, you will see the module listed in grey in the tree. To hide the module, click on the highlighted icon below.

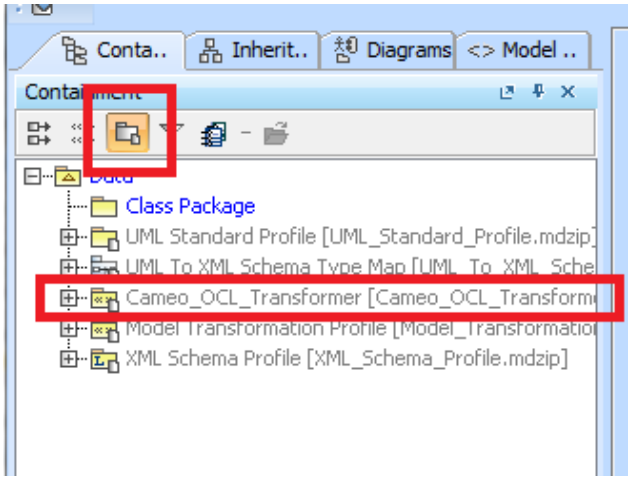

Figure 14 Cameo_OCL_Transformer Module shown in Tree

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

3.2 Generate an XML schema for the Class Model

You will need to generate an XML Schema 'implementation' data model from the class model. The OCL ruleset will be compiled against this implementation data model, and can then be executed against XML data that conforms to this XML schema.

Prerequisite: Cameo Data Modeller must be installed.

To generate an XML schema from a class model, you will need to:

- Transform the UML class model to an XML Schema Model
- Use an XML Schema Code Engineering set to generate the XML Schema from the XML Schema model

For instructions on how to generate an XML Schema, take a look at http://nomos-software.com/blog.

3.3 Configure the Class Model Package

You will need to:

- Configure the package element that contains the class model

3.3.1 Configuring class model package element

- Double click the class model package element in the MagicDraw containment tree
- A Specification dialog opens. Set the Applied Stereotype to 'oclTransformerPackage'.

Two new properties will be listed in the dialog: Ruleset and XML Schema.

Note: If the OCLTransformerPackage stereotype is not available, check that you have included the Cameo_OCL_Transformer module in the project.

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

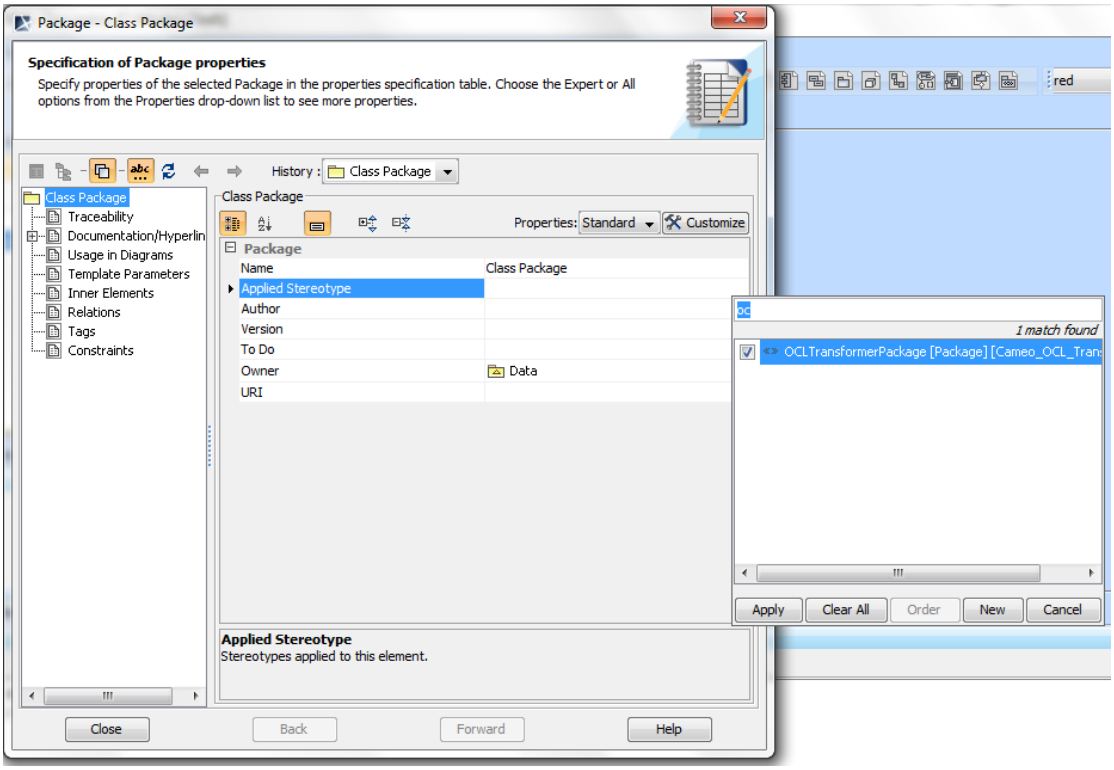

Figure 15 Stereotyping a Package

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

You must provide values for the two new properties:

- Ruleset
- XMLSchema

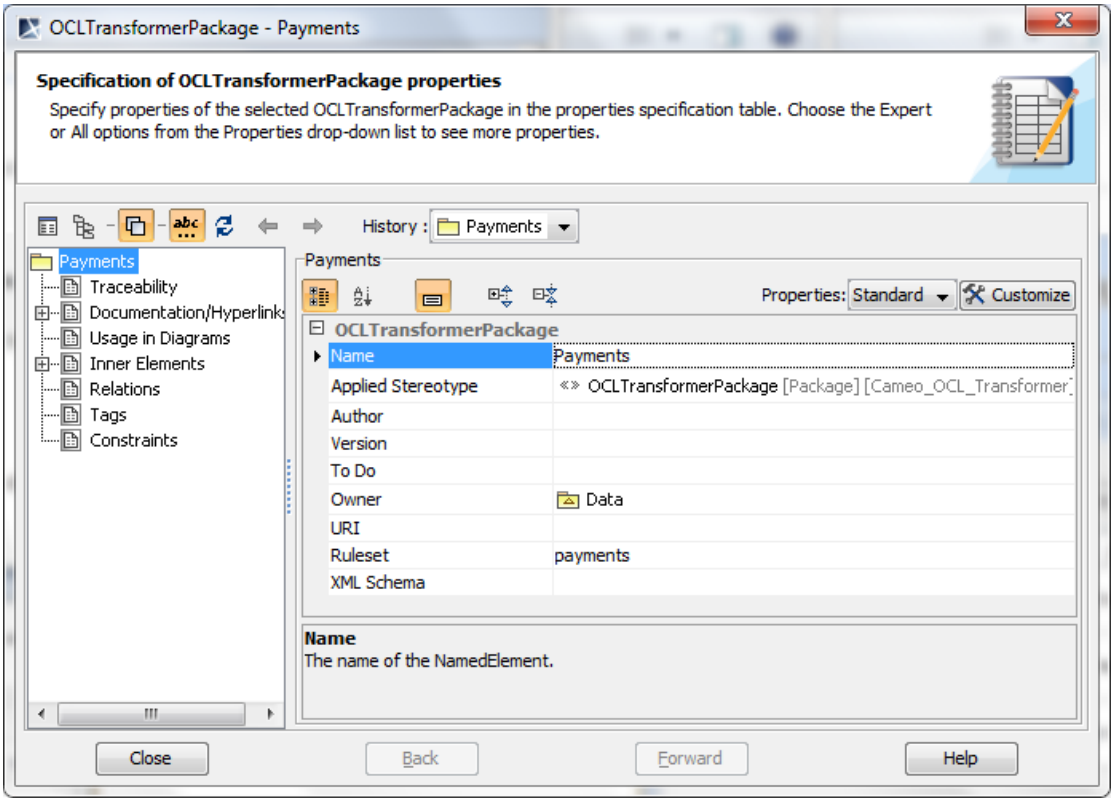

Figure 16 Setting the Ruleset and XMLSchema Properties

Assign a ruleset name to the Ruleset package property. The OCL transformer will generate a ruleset containing all rules found in this package with the ruleset name as this.

Rules are compiled against the XML Schema generated from the class model in the package.

Assign the XML Schema name to the XML Schema property. Location can be defined by a path relative to the location of the current project, or by an absolute path.

If no XML schema name is defined, the Cameo OCL Transformer will by default work schemas located in the project directory.

You can also assign a schema directory to this property. This is useful if schemas are spread across multiple directories. Choose the root directory of the full set of related schemas.

3.4 Adding OCL Rules to Model

To add OCL rules to a MagicDraw model:

- Select the context for the rule (a MagicDraw class)
- Add a constraint element
- Stereotype as an OCLTransformerValidationRule
- Then add to the constraint
    - A name
    - An error message

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

- A ruleset name – ***must be set to ruleset name in the class model package***
- The OCL, tagged as OCL2.0

Here's what OCL rules look like in the containment tree:

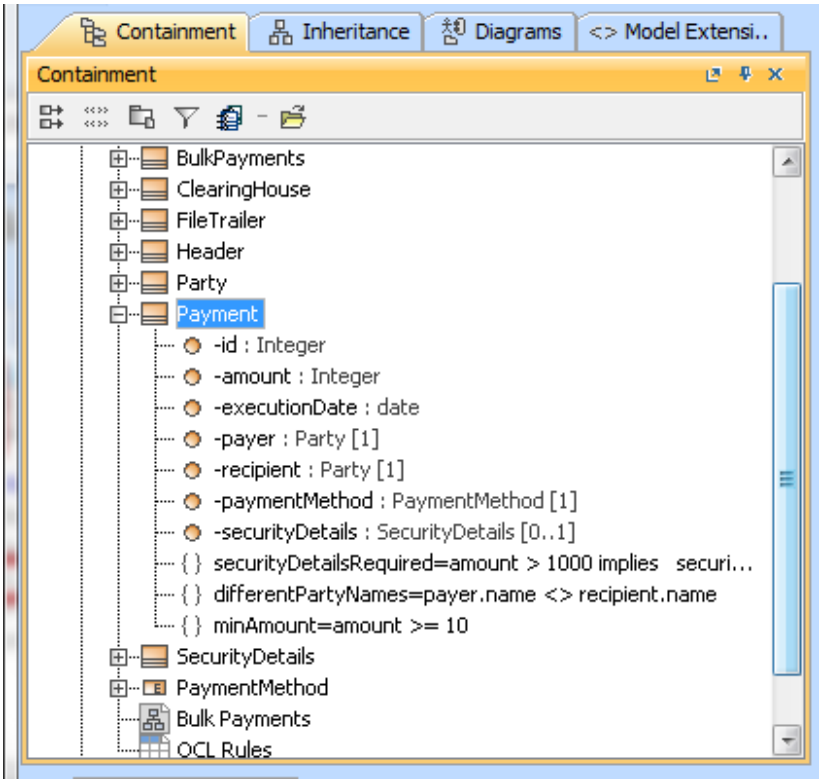

Figure 17 OCL Rules in Tree

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

And here's what an individual OCL rule looks like in the MagicDraw constraint dialog:

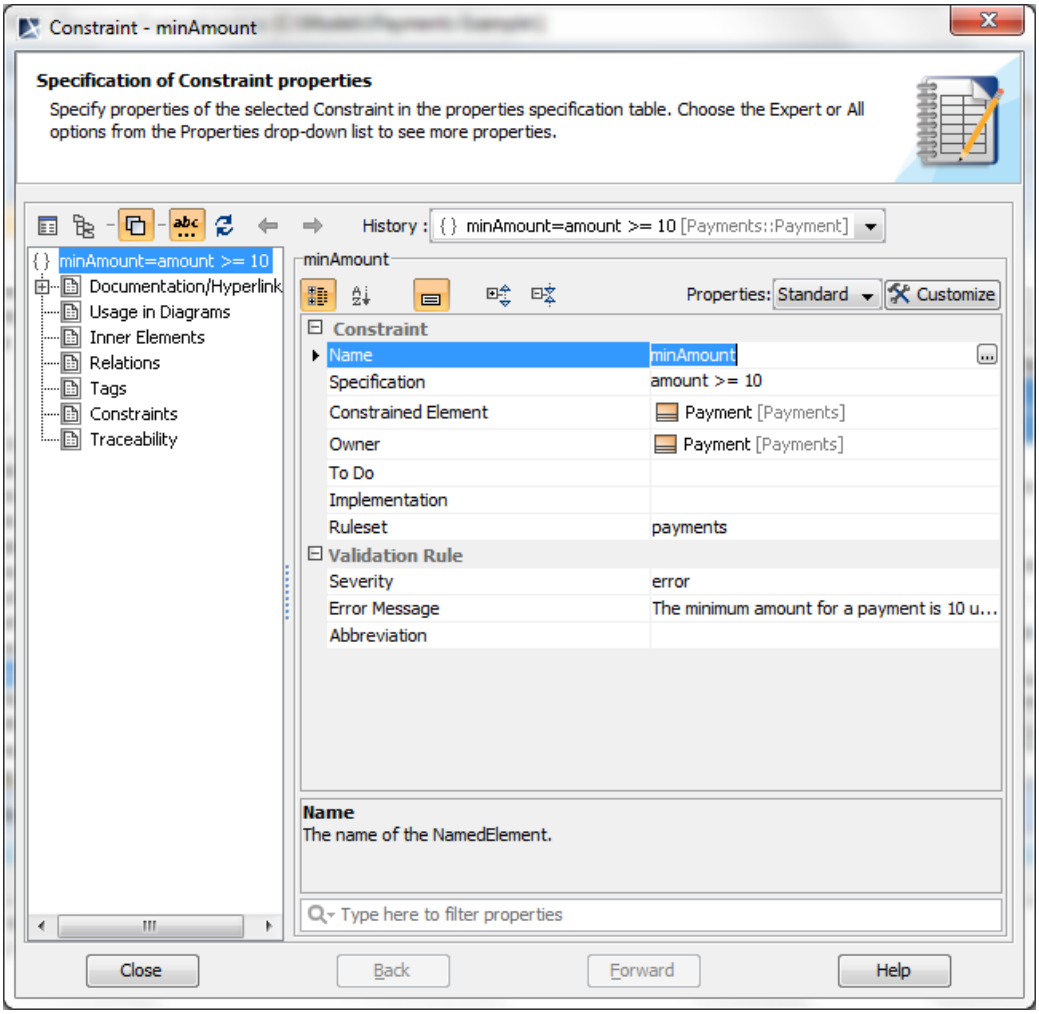

Figure 18 OCL Rule Dialog

For more details and screenshots showing how to add OCL rules to MagicDraw, see the Nomos blog post, Adding OCL rules to MagicDraw:

http://nomos-software.com/blog/how-to-add-ocl-rules-in-magicdraw.

TIP: Once you have created one rule, you can copy and paste this rule. The copied rule will be stereotyped correctly, will be defined as OCL2.0 and will have the ruleset name set.

TIP: Generic Tables are a great way of viewing rules. Set one up for rules using Diagrams->Diagram Wizards->Generic Table Wizard… For details, check out nomos-software.com/blog.

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

3.5 Generate a Ruleset

Select the class model package, right click and select Cameo OCL Transformer->Generate Ruleset.

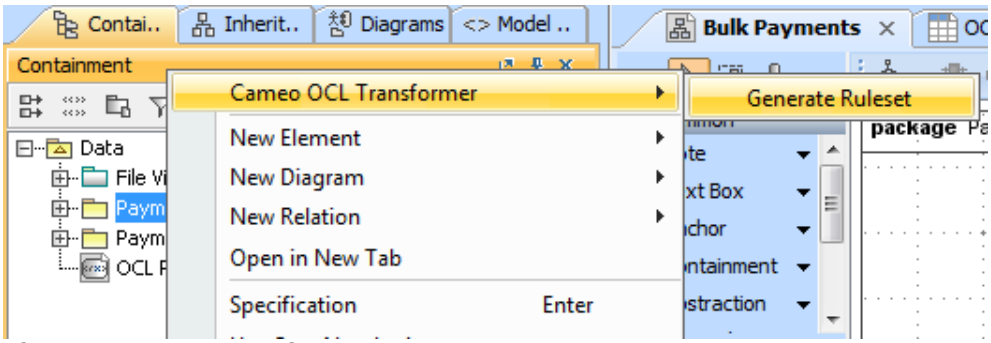

Figure 19 Generate Ruleset

If the menu option isn't available, check that the package has been stereotyped as an OCLTransformerPackage.

This step will compile all OCL2.0 rules in the class package with the same ruleset name as that defined in the package. Any errors detected are displayed in a Validation Results pane. You can locate the rule in error by selecting the error. On the popup menu, choose 'Select in Containment Tree'. Double click the rule, and correct the error.

TIP: You can regenerate the ruleset easily from the Validation Results pane by clicking on the Refresh icon.

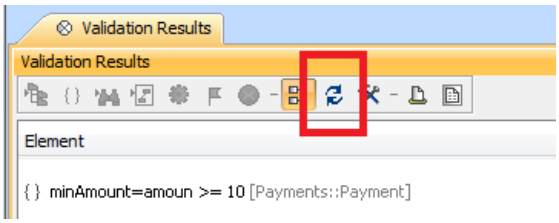

Figure 20 Regenerate Ruleset

The ruleset is only generated once all compilation errors in the OCL rules have been resolved. If the ruleset has already been generated, it will be overwritten.

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

4 Executing Rulesets via MagicDraw

You can execute the generated rulesets against XML files from within MagicDraw using the 'Validate XML' option. As soon as you write a rule, it is good to generate code, execute it and make sure that the rule is working correctly.

The option to Validate XML is always disabled when you start MagicDraw. Once you generate a ruleset, the option becomes enabled.

4.1.1 Create Example XML Files

To execute the rulesets, create some example XML files. We recommend creating example files that exercise the rules you have written. That is, create positive and negative tests for each rule.

Standard XML editing tools can be used to generate sample XML files from the XML schema you have generated.

4.1.2 Validate XML

Select Tools->Cameo OCL Transformer->Validate XML…

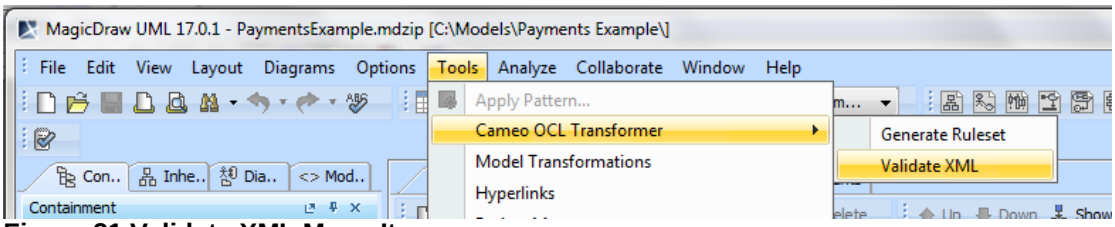

Figure 21 Validate XML Menu Item

You will be prompted to select an XML file

- Browse and select one of the example XML files

You will be prompted to select the directory containing the generated rules. The directory name is the same as the name of the project.

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

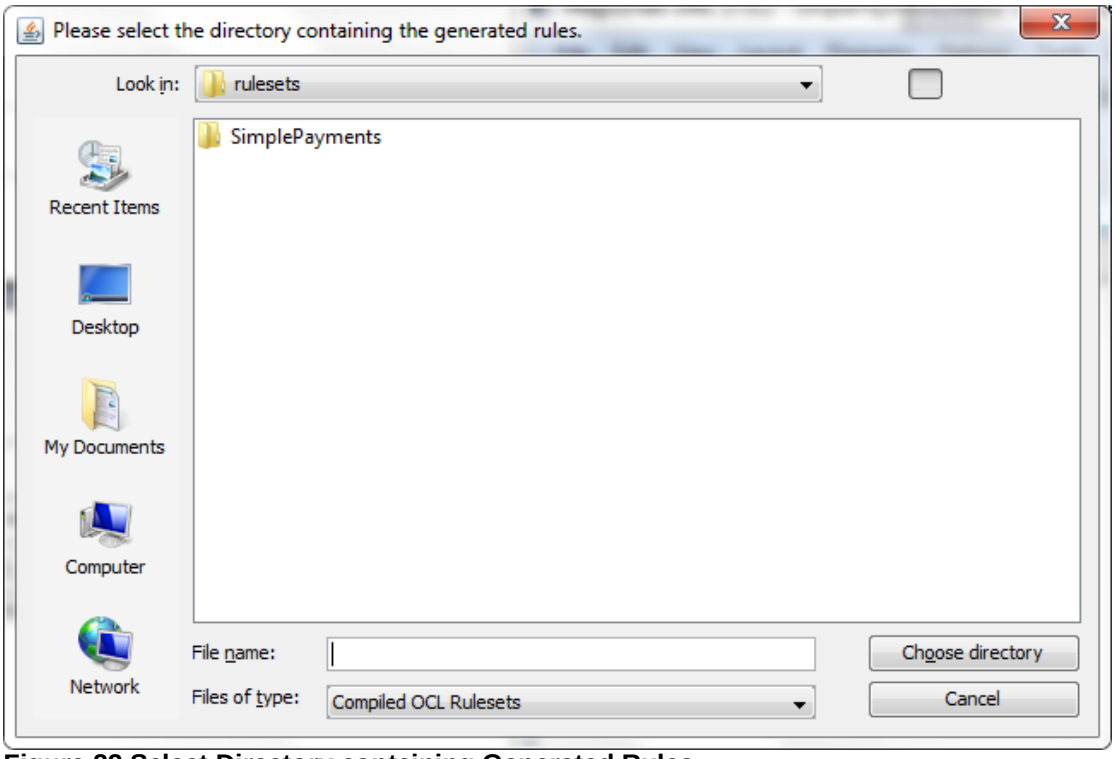

Figure 22 Select Directory containing Generated Rules

The XML file will be validated against the generated ruleset.

Each of the rules that fails is listed in the 'Validation Results' area.

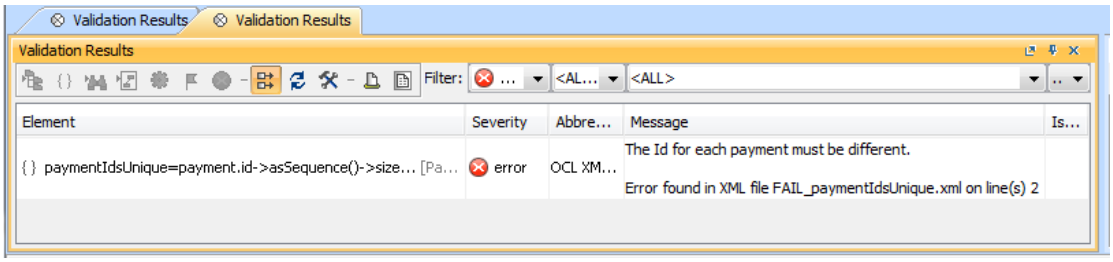

Figure 23 'Validate XML' Results

This is the mechanism for validating XML files against the ruleset through the MagicDraw user interface. The ruleset can be also be called programmatically, either by making calls directly to the ruleset jar (java archive), or by using the Nomos Validation Web Service. For details on calling directly to the ruleset jar, see the document "Nomos Software Using the Runtime Java API.doc". This can be found in:

<MagicDraw_install_directory>/plugins/com.nomagic.magicdraw.cameoocltransformer

5 Learn More

Check out blog posts at nomos-software.com/blog.

Contact Nomos Software to find out about additional features:

- Validation Web Service for executing rulesets

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012

- OCL Extensions
- OCL Queries (used to pinpoint error locations)
- Multiple rulesets
- Support for non-standard UML to XML Schema mappings
- Support for non-XML based data models

6 Contacting Nomos Software Support

Nomos Software support can be contacted by emailing support@nomos-software.com.

© Copyright NOMOSÉIRE Limited trading as Nomos Software 2012